



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/046,389	01/16/2002	Kenneth A. Berger	014600-0003 (B69393)	1678
33649	7590	11/01/2005		
Mr. Christopher John Rourk GODWIN GRUBER, LLP 1201 Elm Street, Renaissance Tower DALLAS, TX 75270			EXAMINER RUTTEN, JAMES D	
			ART UNIT 2192	PAPER NUMBER

DATE MAILED: 11/01/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/046,389	BERGER ET AL.	
	Examiner	Art Unit	
	J. Derek Rutten	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 01 August 2005.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-20 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-20 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|-------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date <u>8/3/05</u> | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is in response to Applicant's amendment dated 14 April 2005, responding to the 14 January 2005 Office action provided in the rejection of claims 1-20, wherein no claims have been amended, no claims have been canceled, and no new claims have been added. The response to the *Notice of Non-Compliant Amendment* (mailed 26 July 2005) was received 01 August 2005. Claims 1-20 remain pending in the application and have been fully considered by the examiner.

2. Applicant's arguments, see paragraph 2 on page 10, filed 14 April 2005, with respect to the rejection(s) of claim(s) 4 and 5 under 35 USC 103(a) have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of prior art of record US 5950001 A to Hamilton et al.

Response to Amendment

3. In an amendment to the specification, appearing on page 2 of the response filed 15 April 2005, there is an instruction to "replace all references from attorney docket number "014600-0003 (B69393) to - 14039-0004 -". However, this does not conform to 37 CFR 1.121(b) which states:

Amendments to the specification, other than the claims, computer listings (§ 1.96) and sequence listings (§ 1.825), must be made by adding, deleting or replacing a paragraph, by replacing a section, or by a substitute specification...

This section further requires that such amendments be made by an instruction that unambiguously identifies the location of such amendments. Applicant's amendment does not

Art Unit: 2192

replace a paragraph or section, and does not unambiguously identify the location of the amendment. Clarification is required.

4. Applicant's amendment responding to the *Notice of Non-Compliant Amendment* has been received. In that response, Applicant indicated that no difference was found between the original claim 15 and the claim 15 as submitted on 4/14/05, but that a difference was found in claim 14. While the Examiner gratefully acknowledges the identification and correction of claim 14, claim 15 remains uncorrected. The first line on page 5 of the response submitted 8/1/05 reads "from one for more classes", while the original claim 15 reads -from one or more classes-. In light of Applicants remarks, this claim has been interpreted in terms of the original claim language. Correction is required.

Response to Arguments

5. In regard to the rejection of claim 1 under 35 U.S.C. 112, Applicant has argued on page 7 that the term "operator" is not unclear since the specification provides no description of mathematical operators, but does describe users and programmers. However, the specification does in fact describe programming and programming languages in general, which draw heavily upon mathematics and commonly use operators to assemble and perform data structure manipulation. Further, review of the specification did not reveal use of the term "operator" to describe a user, programmer, or developer. The unqualified use of the term "operator" in the context of computer programming remains unclear. Therefore, this argument is not convincing.

Art Unit: 2192

6. Applicant's arguments, see pages 7 and 8, filed 4/14/05, with respect to the rejection of claim 15 under 35 USC 112 second paragraph have been fully considered and are persuasive.

This rejection of claim 15 has been withdrawn.

7. In response to applicant's argument on pages 8 and 9 in reference to the rejection of claim 1 under 35 U.S.C. 102, that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., "claim 1 eliminates manual programming" – page 9 line 9) are not recited in the rejected claims. Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Also, in regard to claim 1, it is noted that Applicant admits that "users select from a predetermined set of interface features" (page 9 lines 11-12). In this context it is not clear what the Applicant is arguing, since the selection of features by a user represents manual programming.

8. Applicant argues at the bottom of page 9 in reference to the rejection of claim 2 under 35 U.S.C. 102 that Chiang does not disclose a developer user interface class system that operates in conjunction with the user interface code to provide modified user interface features. However, this limitation is provided by Chiang in FIG. 5 and on page 6 paragraph [0073] which discloses the use of such a developer user interface class system by the developers of FIG. 2 element 210, to provide modified user interface features. Thus, this argument is not convincing.

Art Unit: 2192

9. Applicant essentially argues at the top of page 10 in reference to the rejection of claim 3, that Chiang does not disclose a user interface class system that comprises a developer handler class system. After further review, this claim is believed to have issues under 35 U.S.C. 112 1st paragraph, as explained in the *Claim Rejections - 35 USC § 112* section below. However, a response to certain aspects of Applicant's arguments is warranted. Applicant asserts the following as a basis for the argument:

The Examiner has identified the identical structure for the handler class system and the developer handler class system, which is a clear indication that the Examiner's construction of these terms is incorrect. Not only is that construction contrary to the **definitions in the specification**, it is also contrary to the **plain meaning**, which requires items that have two different names to be two different things. (Emphasis added)

It is noted that the Applicant has neither provided any specific citations that provide definitions in the specification, nor provided any interpretations of plain meaning. Applicant has not particularly pointed out how the language of the claims distinguishes it from the reference. Nonetheless, while the Office Action does provide similar citations in response to limitations appearing in claim 1 as well as in claim 3, more detailed descriptions of both are found in Chiang's disclosure. See page 5, paragraph [0067]:

Event handler code 620 comprises a skeleton or framework made up of an event handler (click method) object that accepts requests from HTML <input> tags with JLMClick name and submit type attributes in the HTML files. As discussed above in connection with FIG. 7, for each JLMClick submit <input> tag identified by web application generator 205, **a corresponding event handler (click method) is generated** to handle the request. Upon receipt of event handler code 620 as an output from web application generator 205, **web developers 215 simply need to write appropriate source code into the event handler code skeleton** to drive the application business logic in response to user interactions. (Emphasis added)

As is clearly evident in the above paragraph, a developer handler system develops handler code based on the event handler code generated by the web application developer. Applicant continues as follows:

Art Unit: 2192

This makes no sense, unless the Examiner is merely construing the user interface states to be the same as the modified user interface states in order to avoid having to find a reference that shows both.

As the above citation of Chiang paragraph [0067] shows, the event handler code skeletons are modified “to drive the application business logic in response to user interactions.” Such “user interactions” occur through a user interface, which in conjunction with the modified handler code provides modified user interface states. Contrary to Applicant’s assertion, rather than avoiding finding a reference that shows both, the Examiner has found both in Chiang.

10. Applicant’s arguments, see page 10 paragraph 2 and page 12 paragraph 2, with respect to the rejections of claims 4, 5, 11, and 12, respectively, have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground of rejection is made in view of Chiang and prior art of record US 5950001 A to Hamilton et al.

11. In the last paragraph on page 10 following on to page 11, and page 12 paragraph 3, Applicant argues that Chiang does not disclose all the limitations of claims 6, 7, 13, and 14, respectively. Claim 6 recite a list of classes prefaced by the phrase “the user interface class includes one or more of the group comprising...” As is clear from this passage, the list of classes is presented using the alternative phrase “one or more”. In this case, disclosure by Chiang of one of the classes provides coverage of the claim. Claims 7, 13, and 14 use similar alternative phrasing and are likewise covered.

Art Unit: 2192

12. In paragraph 2 on page 11, Applicant argues that Chiang requires manual processes, while claim 8 does not. It is noted that the features upon which applicant relies (i.e., “No manual processes are required...”) are not recited in the rejected claim. Applicant argues similarly with respect to claims 9 and 10. Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

13. On pages 12 –13, Applicant provides several arguments suggesting that the rejection of claim 15 is improper.

The first argument (page 12 paragraph 4) suggests that Chiang fails to disclose a primary code generator receiving one or more user selections from one or more classes and generating primary code. However, further review of Chiang reveals such a primary code generator. See page 2 paragraph [0010]:

In accordance with one embodiment of the web application generator, there is provided a method of generating computer code for a web application, and dynamically binding input files from graphic designers and source code from web developers, comprising the web application server **receiving input files from graphic designers or business analysts, wherein the input files are at least one web application graphical user interface.** The web application server determines if an application framework code is available for the web application, and retrieves the application framework code from an application directory. **If the application framework code is not available for the web application, then the web application server generates the application framework code, along with a business logic foundation code, an event handler skeleton and a graphical user interface code.** (Emphasis added)

The web application generator receives user selections in the input files that are used to generate primary code.

The next argument (page 12 paragraph 5) suggests that Chiang fails to disclose a developer code generator receiving the primary software code and generating developer software code. These features are disclosed in Chiang. See Chiang page 2 paragraph [0010]:

Further, based on work by web developers, the web application server receives web application business logic objects and event handlers from the web developers, and **organizes the application framework code, web application business logic objects and event handler methods into web application source code**. The method further comprises compiling the web application source code. Modified input files may then be received by the web application server from the graphic designers or business analysts, and the modified input files are compiled and dynamically bound with the compiled web application source code at runtime.

Also in paragraph 5 on page 12, Applicant points out in bold underlined text that the same section of the reference (paragraph [0010]) was used to refer to two elements of the claim, and suggests that this “eliminates an entire element of the claim”. While the rejection does cite the same section of the reference, to suggest that this eliminates an entire element of the claim is a mischaracterization of the reference. Paragraph [0010] of Chiang discusses a broad summary of the invention and recounts many of the fundamental features, including code generators, graphical user interfaces, framework code, foundation code, modified input files, event handler generation, etc. Thus, numerous features of Chiang’s invention are disclosed in “the same section” of the reference. It is not unreasonable to suggest that Chiang can disclose more than one element of the invention in a relatively long paragraph. Thus, this argument is not convincing.

In the last paragraph on page 12 continuing on page 13, Applicant argues that Chiang’s editing process would not provide code that is backwards and forwards compatible with other tiers in a multi-tier architecture. It is noted that the language of claim 15 does not expressly refer to mutually compatible “tiers”, layers, or other such terminology. Applicant further argues that Chiang’s iterative modification makes it “impossible for the primary code to be modified and for those modifications to be compatible with a version of the code that has been iteratively modified. However, this argument is not entirely convincing. One of ordinary skill in the art would recognize that as long as the interface, or API, of a code base were left unchanged, the

Art Unit: 2192

implementation of that code could be modified extensively while remaining compatible with code that depends upon it. Further, Chiang meets the language of the claim, which simply requires modifications to be compatible. Since the modified files can be compiled and dynamically bound as indicated in Chiang page 2 paragraph [0010] (see previous Office Action), they *must* be compatible, otherwise a compiler would produce errors.

In the second paragraph on page 13, Applicant essentially argues that Chiang neither discloses a primary code editor that modifies classes, nor compatibility of modified code with prior code. However, Chiang discusses the generation of class files. See page 5 paragraphs [0062] and [0063]. If developers wish to develop application code based on these generated class files, they must use an editor capable of editing classes. Furthermore, the plain language of the claims does not place any limitations of the type of “class” files that are used. For example, they are not restricted to being object-oriented class files. Rather, broad interpretation of the term “class” permits the interpretation of Chiang’s input files as being such class files. In regard to compatibility, Applicant argues that it is impossible under Chiang. This argument is wholly unpersuasive. As discussed earlier, as long as an API is maintained, the implementation of such code can be modified dramatically while the interface remains compatible. Far from being impossible, this practice is quite commonplace.

Specification

14. The disclosure is objected to because of the following informalities: Paragraph [0024] of the originally filed specification makes reference to “U.S. Patent 09/689,067” in line 10 on page

Art Unit: 2192

8, which should instead be --U.S. Patent Application 09/689,067-- since it refers to an application which is now abandoned. Appropriate correction is required.

Claim Rejections - 35 USC § 112

15. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

16. Claim 3 is rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. Claim 3 recites "...wherein the user interface class system further comprises a developer handler class system..." This feature is not supported by the originally filed specification. Inspection of Figure 1 shows a code generator system 102 which includes primary, developer, and site-specific code generators 104, 106, and 108, respectively. Each code generator includes a user interface class system, and a handler class system. The user interface class system and the handler class system are separated in the figure, and associated text in the specification does not provide any description where a user interface class system comprises a handler class system. While paragraph [0022] provides support for a handler class system 112 that is coupled to a user interface class system 110, this would not enable one of ordinary skill in the art to make and/or use the invention of claim 3.

17. Claims 8 and 15 are rejected under 35 U.S.C. 112, first paragraph, because the specification, while being enabling for a code generator that receives user selections *for* classes

(page 6 lines 23-27 and 35-36, page 7 lines 23-28, etc.), does not reasonably provide enablement for a code generator that receives user selections *from* classes. The specification does not enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the invention commensurate in scope with these claims. Interpretation of the phrase “receiving a selection of a user interface feature from a user interface class” (line 2 of claim 8) will be made to instead mean --receiving a selection of a user interface feature for a user interface class--. Likewise, the phrase “receiving one or more user selections from one or more classes” (Claim 15 appearing on page 5 line 1) will be made to instead mean --receiving one or more user selections for one or more classes--.

18. Claim 15 is rejected under 35 U.S.C. 112, first paragraph, because the specification, while being enabling for a primary code editor that modifies primary software code (see page 35, paragraph [0086] lines 14-23), does not reasonably provide enablement for a primary code editor that modifies developer classes. The specification does not enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the invention commensurate in scope with these claims. Interpretation of the phrase “one or more of the classes” (appearing on page 5 line 5) can be made to refer to both “one or more classes” (page 5 line 1) and “one or more developer classes” (page 5 line 4). The specification does not discuss a primary code editor that edits developer classes.

19. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Art Unit: 2192

20. Claims 1-7 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

21. Claim 1 contains the phrase "by an operator" in line 6. It is not clear if the word operator is referring to a user of the system, or if it refers to an operator of the code that links the two features in some way (e.g. the '+' operator in '5+2'). For the purpose of further examination, this will be interpreted to refer to --a user of the system--.

22. Claim 1 recites the limitation "the selected user interface features" in lines 8-9. There is insufficient antecedent basis for this limitation in the claim. This limitation will be interpreted as --the assembled user interface features--.

23. Claims 1-7 are rejected under 35 U.S.C. 112, second paragraph, as failing to set forth the subject matter which applicant(s) regard as their invention. Evidence that claims 1-20 fail(s) to correspond in scope with that which applicant(s) regard as the invention can be found in the reply filed 14 April 2005. In that paper, applicant has stated "claim 1 eliminates manual programming" (page 9 line 9) and this statement indicates that the invention is different from what is defined in the claim(s) because the claim calls for "features that can be assembled into a user interface by an operator". If the term "operator" is interpreted to be a user of the system, then manual programming is present in the claim.

Art Unit: 2192

24. Claims 2-7 are rejected for being dependent upon a rejected base claim.

Claim Rejections - 35 USC § 102

25. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

26. Claims 1-3, 5-10, 12-15, 17, 18 and 20 are rejected under 35 U.S.C. 102(a) as being anticipated by U.S. Patent Application Publication US 2001/0037490 A1 by Chiang (hereinafter “Chiang”).

In regard to claim 1, Chiang discloses:

A system for generating user interface code (See FIG. 2) comprising:

a user interface class system generating user interface class code, wherein the user interface class code includes two more user interface features that can be assembled into a user interface by an operator; See paragraph [0037]:

As shown in FIG. 5, the first step comprises one or multiple graphic designers and/or business analysts 210 creating web application screens using visual editors or a text editor (step 505). In one embodiment, the application screens are written in HTML format using a visual HTML editor, such as Microsoft FrontPage, Adobe GoLive or Netscape Composer.

Also see paragraph [0052]:

As illustrated in FIG. 6, the web application generator 205 reads the set of web application screens as the input 605 and generates web application source code 610 as the output.

Also paragraph [0062]

In one embodiment, web application source code output 610 is generated as a corresponding Java class.

Also see paragraph [0095].

a handler class system generating handler class code, wherein the handler class code includes one or more states for each user interface feature of the user interface class. See page 3 paragraph [0026] and FIG. 2 as cited above; also paragraph [0035] and FIG. 4 element 415: "Event Handler". Also paragraph [0057]:

Ultimately, based on the input tag, attribute name(s) and associated attribute value(s), web application generator 205 relies on a particular "rule" (or fixed formula) within web application server memory 325 to generate event handler code 620 and GUI code 625 (step 730).

Chiang's handler code controls the web application in order to transition from one state to another.

wherein the user interface class and the handler class cause the selected user interface features and associated states for the user interface features be generated when the user interface code is executed. See page 6 paragraph [0078]:

Based on a request for the web application from a web browser, the web application server 100 dynamically binds graphical user interface (GUI) files with web application business logic objects at runtime in order to provide the web application to the web browser (step 535).

In regard to claim 2, the above rejection of claim 1 is incorporated. Chiang further discloses a developer user interface class system. See FIG. 2 element 210 and FIG. 4 element 405.

In regard to claim 3, the above rejection of claim 1 is incorporated. Chiang further discloses a developer handler class system. See FIG. 2 element 215 and FIG. 4 element 415.

In regard to claim 5, the above rejection of claim 3 is incorporated. Chiang further discloses a site-specific handler class system that generates a site-specific handler class through the customization of a properties file. See page 5 paragraph [0071].

In regard to claim 6, the above rejection of claim 1 is incorporated. Chiang further discloses a report user interface class on page 7 paragraph [0097].

In regard to claim 7, the above rejection of claim 1 is incorporated. Chiang further discloses a report handler class on page 7 paragraph [0098].

In regard to claim 8, Chiang discloses:

A method for generating user interface code comprising:

receiving a selection of a user interface feature from a user interface class; See page 2 paragraph [0010]:

Modified input files may then be received by the web application server from the graphic designers or business analysts, and the modified input files are compiled and dynamically bound with the compiled web application source code at runtime.

retrieving a handler associated with the user interface feature that includes one or more states; See page 2 paragraph [0010]:

Further, based on work by web developers, the web application server **receives** web application business logic objects and **event handlers** from the web developers, and organizes the application framework code, web application business logic objects and event handler methods into web application source code.

generating one or more code elements that cause a user display to be generated when executed that includes the user interface feature having the one or more states. See paragraph [0010] starting on page 1:

In accordance with one embodiment of the web application generator, there is provided a method of **generating computer code** for a web application, and dynamically binding input files from graphic designers and source code from web developers, comprising the web application server receiving input files from graphic designers or business analysts, wherein the input files are at least one web application **graphical user interface**.

In regard to claims 9 and 10, the above rejection of claim 8 is incorporated. All further limitations have been addressed in the above rejection of claims 2 and 3, respectively.

In regard to claims 12-14, the above rejection of claim 8 is incorporated. All further limitations have been addressed in the above rejections of claims 5-7, respectively.

In regard to claim 15, Chiang discloses:

A system for generating software code comprising:

a primary code generator receiving one or more user selections for one or more classes and generating primary software code; See page 2 paragraph [0010]:

In accordance with one embodiment of the web application generator, there is provided a method of **generating computer code** for a web application, and dynamically binding input files from graphic designers and source code from web developers, comprising the web application server receiving input files from graphic designers or business analysts, wherein the input files are at least one web application **graphical user interface**.

a developer code generator receiving the primary software code and one or more user selections from one or more developer classes and generating developer software code; primary code editor modifying said one or more of the classes; See page 2 paragraph [0010]:

Further, based on work by web developers, the web application server receives web application business logic objects and event handlers from the web developers, and **organizes the application framework code, web application business logic objects and event handler methods into web application source code.**

Also see page 5 paragraph [0069]:

Upon receipt of business logic foundation code 630 as an output from web application generator 205, web developers 215 assemble reusable object-oriented business components and write all the core business logic that drives the web application, and particularly its function.

An editor is inherently required in order to assemble and modify components and logic; otherwise any output would be the same as the input.

wherein the modifications made to said one or more classes by the primary code editor result in the generation of code that is compatible with the developer software code. See page 6 paragraph [0075]:

When web developers 215 are finished preparing web application source code 610, web application source code 610 is either compiled or interpreted by the programming language compiler/interpreter in which the code is written (step 530).

Compatibility is inherently required for compilation otherwise compilation would fail.

In regard to claim 17, the above rejection of claim 15 is incorporated. All further limitations have been addressed in the above rejection of claims 1 and 8.

In regard to claim 18, the above rejection of claim 15 is incorporated. All further limitations have been addressed in the above rejection of claims 2 and 3.

In regard to claim 20, the above rejection of claim 15 is incorporated. All further limitations have been addressed in the above rejection of claims 1 and 8.

Claim Rejections - 35 USC § 103

27. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

28. Claims 4, 11, 16, and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chiang as applied to claims 2, 3, 5, 8, 9, 12, and 15 above, in view of US 5,950,001 to Hamilton et al. (hereinafter "Hamilton").

In regard to claim 4, the above rejection of claim 2 is incorporated. Chiang further discloses site-specific modification of a user interface and customization for specific user sites (See page 5 paragraph [0071]). Chiang does not expressly disclose site-specific user interface features. However, Chiang discloses modification and dynamic binding of modified code in order for a user interface to be customized (See page 2 paragraph [0010]). Further, Hamilton teaches that site-specific features of a user interface class can be configured to operate with existing classes through the use of a

Art Unit: 2192

“customizer” (See column 2 lines 17-24). It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Hamilton’s customizer with Chiang’s class system. One of ordinary skill would have been motivated to provide specific modifications according to the needs of a particular user (Hamilton column 1 lines 20-23).

In regard to claim 11, the above rejection of claims 8 and 9 are incorporated. All further limitations have been addressed in the above rejection of claim 4.

In regard to claims 16 and 19, the above rejection of claim 15 is incorporated. All further limitations have been addressed in the above rejection of claims 4 and 5, respectively.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to J. Derek Rutten whose telephone number is (571) 272-3703. The examiner can normally be reached on T-F 6:00 - 4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner’s supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2192

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

jdr

A handwritten signature in black ink, appearing to read 'Tuan Dam', with a long horizontal stroke extending to the left.

TUAN DAM
SUPERVISORY PATENT EXAMINER